

# Approximation-Degree-Based Interpolation: A New Interpolation Method

Shiyu Lian

Xi'an Shiyou University, Xi'an, China

sylian@xsyu.edu.cn

## Abstract

This paper introduces the measure of approximate-degree and the concept of approximate-degree function between numerical values, thus developing a new interpolation method — approximation-degree-based interpolation, i.e., AD interpolation. One-dimensional AD interpolation is done directly by using correlative interpolation formulas;  $n(n>1)$ -dimensional AD interpolation is firstly separated into  $n$  parallel one-dimensional AD interpolation computations to do respectively, and then got results are synthesized by Sum-Times-Difference formula into a value as the result value of the  $n$ -dimensional interpolation. If the parallel processing is used, the efficiency of  $n$ -dimensional AD interpolation is almost the same as that of the one-dimensional AD interpolation. Thus it starts a feasible and convenient approach and provides an effective method for high-dimensional interpolations. Furthermore, if AD interpolation is introduced into machine learning, a new instance-based learning method is expected to be realized.

**Keywords** Approximation-Degree; High-Dimensional Interpolation; First Separating then Synthesizing; Parallel Processing; Sum-Times-Difference Formula; Instance-Based Learning

## 1 Introduction

With the rapid development of artificial intelligence and machine learning, some of the traditional numerical computation methods such as interpolation, data fitting and regression analysis have aroused people's interest and become active fields again. Although there has been sufficient research on interpolation, the results of high-dimensional interpolation are still relatively few, and some of the existing high-dimensional interpolation methods<sup>[2]</sup> have a large load of computation and are not suitable for local interpolation. High-dimensional interpolation has important applications in artificial intelligence and machine learning. For example, instance-based learning<sup>[3,4]</sup> often involves high-dimensional interpolation. Therefore, the high-dimensional interpolation, especially that suitable for local interpolation, is still an

important subject worthy of serious study.

Inspired by the approximate evaluation method of flexible linguistic functions<sup>[5, 6]</sup> in reference [1], this paper intends to introduce the measure of approximate-degree between numerical values to study the approximate evaluation of numerical functions, and then explores the approach and method of interpolation based on approximate-degree, especially, the method of high-dimensional interpolation based on approximate-degree.

## 2 Approximation Axiom, Approximation-Degree, and Approximation-Degree Transmission

**Approximation Axiom** Let  $\mathbf{R}$  be real number field,  $\mathbf{R}^n$  be  $n$ -dimensional real vector space,  $U=[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \subset \mathbf{R}^n$  ( $n \geq 1$ ),  $V=[c, d] \subset \mathbf{R}$ , and  $y=f(\mathbf{x})$  ( $\mathbf{x}=(x_1, x_2, \dots, x_n)$ ) be a continuous and non-chaotic function relation from  $U$  to  $V$ . If  $y_0=f(\mathbf{x}_0)$  ( $\mathbf{x}_0 \in U$ ) is known, then when  $\mathbf{x} \in U$  is approximate to  $\mathbf{x}_0$ ,  $y=f(\mathbf{x}) \in V$  is also approximate to  $y_0$ .

**Definition 2-1** Let  $\mathbf{R}$  be real number field,  $x_0 \in [a, b] \subset \mathbf{R}$ , and  $[\alpha_0, \beta_0] \subset [a, b]$  be a neighborhood of  $x_0$ , which is called the approximation domain of  $x_0$ . For  $\forall x \in [a, b]$ , say  $x$  is approximate to  $x_0$  if and only if  $x \in [\alpha_0, \beta_0]$ .

**Definition 2-2** Let  $\mathbf{R}^n$  be  $n$ -dimensional real vector space,  $U$  be an  $n$ -dimensional subspace of  $\mathbf{R}^n$ ,  $\mathbf{x}_0=(x_{1_0}, x_{2_0}, \dots, x_{n_0}) \in U$ , and  $C \subset U$  be a "circular" region with center  $\mathbf{x}_0$ , which is called an approximation domain of  $\mathbf{x}_0$ . For  $\forall \mathbf{x}=(x_1, x_2, \dots, x_n) \in U$ , say  $\mathbf{x}$  is approximate to  $\mathbf{x}_0$  if and only if  $\mathbf{x} \in C$ .

**Definition 2-3** Let  $\mathbf{R}^n$  be  $n$ -dimensional real vector space,  $U=[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \subset \mathbf{R}^n$ , and  $\mathbf{x}_0=(x_{1_0}, x_{2_0}, \dots, x_{n_0}) \in U$ . For  $\forall \mathbf{x}=(x_1, x_2, \dots, x_n) \in U$ , say  $\mathbf{x}$  is strictly approximate to  $\mathbf{x}_0$  if and only if the components  $x_1, x_2, \dots, x_n$  of  $\mathbf{x}$  are approximate to the components  $x_{1_0}, x_{2_0}, \dots, x_{n_0}$  of  $\mathbf{x}_0$ , respectively, i.e.,  $x_i \in [\alpha_i, \beta_i] \subset [a_i, b_i]$  ( $[\alpha_i, \beta_i]$  is approximation domain of  $x_{i_0}$ ,  $i=1, 2, \dots, n$ ), and the "square" region  $[\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \dots \times [\alpha_n, \beta_n] \subset U$  is called strict approximation domain of  $\mathbf{x}_0$ .

In contrast to the strict approximation and strict approximation domain in Definition 2-3, we call the approximation and approximation domain defined in Definition 2-2 to be ordinary approximation and ordinary approximation domain. It can be seen that the ordinary approximation is the approximation at the level of points (vectors), while the strict approximation is the approximation at the level of the coordinates of points (components of vectors). The relation between the ordinary approximation domain and the strict approximation domain of the same point  $\mathbf{x}_0$  is shown in Figure 2-1. The illustration also shows visually the relationship between the ordinary approximation and the strict approximation.

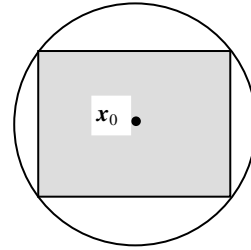


Figure 2-1 Illustration of the relation between ordinary approximation domain and strict approximation domain

We know that the approximate evaluation of a function is to obtain the approximate value of  $f(\mathbf{x}')$  in the condition of a pair  $(\mathbf{x}_0, y_0)$  of corresponding values of function  $y=f(\mathbf{x})$  ( $\mathbf{x}=(x_1, x_2, \dots, x_n)$ ) and  $\mathbf{x}'$  approximate to  $\mathbf{x}_0$  being known. By the

Where the circular region is the ordinary approximation domain of point  $\mathbf{x}_0$ , and the square region is its strict approximation domain.

axiom of approximation above, when  $x'$  is approximate to  $x_0$ ,  $f(x')=y$  is approximate to  $f(x_0)=y_0$ . But approximation is a flexible concept (i.e., flexible linguistic value<sup>[1]</sup>). In order to characterize precisely these two approximation relations and the relation between them, we introduce the measure of approximation-degree.

**Definition 2-4** Let  $\mathbf{R}$  be real number field,  $x_0 \in [a, b] \subset \mathbf{R}$ , and  $[\alpha_0, \beta_0] \subset [a, b]$  be the approximation domain of  $x_0$ . Set

$$A_{x_0}(x) = \begin{cases} 1 - \frac{x_0 - x}{x_0 - \alpha_0} = \frac{x - \alpha_0}{x_0 - \alpha_0}, & x \in [\alpha_0, x_0] \\ 1 - \frac{x - x_0}{\beta_0 - x_0} = \frac{x - \beta_0}{x_0 - \beta_0}, & x \in [x_0, \beta_0] \end{cases} \quad (2-1)$$

to be called the approximation-degree of  $x$  to  $x_0$ . Where  $x_0 - \alpha_0 = rl$  is called left approximation radius of  $x_0$ ,  $\beta_0 - x_0 = rr$  is called right approximation radius of  $x_0$ . We call the function relation defined by the Formula (2-1) to be the approximation-degree function of  $x_0$ .

It can be seen that the range of approximation-degree function  $A_{x_0}(x)$  is  $[0, 1]$  and it is also reversible. In fact, it's easy to obtain that

$$A_{x_0}(x)^{-1} = \begin{cases} d_x(x_0 - \alpha_0) + \alpha_0 \\ d_x(x_0 - \beta_0) + \beta_0 \end{cases} \quad (2-2)$$

where  $d_x \in [0, 1]$  is the approximation-degree of  $x$  to  $x_0$ .

**Definition 2-5** Let  $\mathbf{R}^n$  be  $n$ -dimensional real vector space,  $U$  be an  $n$ -dimensional subspace of  $\mathbf{R}^n$ ,  $\mathbf{x}_0 = (x_{1_0}, x_{2_0}, \dots, x_{n_0}) \in U$ , and  $C \subset U$  be an approximation domain of  $\mathbf{x}_0$ . Set

$$A_{\mathbf{x}_0}(\mathbf{x}) = 1 - \frac{\|\mathbf{x}_0 - \mathbf{x}\|}{r} = 1 - \frac{\sqrt{\sum_{i=1}^n (x_{i_0} - x_i)^2}}{r}, \quad \mathbf{x} = (x_1, x_2, \dots, x_n) \in C \quad (2-3)$$

to be called the approximation-degree of  $\mathbf{x}$  to  $\mathbf{x}_0$ . Where  $r$  is the radius of  $C$ , and which is called the approximation radius of  $\mathbf{x}_0$ .

We further consider the dependence relation between approximation-degree  $A_{y_0}(y)$  and  $A_{x_0}(\mathbf{x})$  under the constraint of function relation  $y = f(\mathbf{x})$ . By Approximation Axiom, the closer  $\mathbf{x}$  is to  $\mathbf{x}_0$ , the closer  $y$  is to  $y_0$ , thus the closer  $A_{y_0}(y)$  is to  $A_{x_0}(\mathbf{x})$ . Then when the corresponding approximation domains are small, we can simply agree that the two are equal, that is, set

$$A_{y_0}(y) = A_{x_0}(\mathbf{x})$$

This equation is equivalent to transmitting the approximation-degree of the independent variable  $\mathbf{x}$  to  $\mathbf{x}_0$  to the corresponding function value  $y$  as the approximation-degree of  $y$  to  $y_0$ .

### 3 Approximate Evaluation of Functions Based on Approximation-Degree

#### 3.1 Approximate Evaluation of Univariate Functions Based on Approximation-Degree

Let  $\mathbf{R}$  be real number field,  $U = [a, b] \subset \mathbf{R}$ ,  $V = [c, d] \subset \mathbf{R}$ ,  $y = f(x)$  be a continuous and non-chaotic function relation from  $U$  to  $V$ . In the condition that a pair  $(x_0, y_0)$  of corresponding values of function  $y = f(x)$  and  $x'$  approximate to  $x_0$  being known, find the approximate value of  $f(x')$ .

Let the approximation domain of  $x_0$  be  $[a_1, b_1] \subset [a, b]$ . According to the definition of the approximation-degree function above, the approximation-degree function of  $x_0$  is

$$A_{x_0}(x) = \begin{cases} \frac{x-a_1}{x_0-a_1}, & x \in [a_1, x_0] \\ \frac{x-b_1}{x_0-b_1}, & x \in [x_0, b_1] \end{cases} \quad (3-1)$$

And let the approximation domain of  $y_0$  is  $[c_1, d_1] \subset [c, d]$ , the approximation-degree function of  $y_0$  is

$$A_{y_0}(y) = \begin{cases} \frac{y-c_1}{y_0-c_1}, & y \in [c_1, y_0] \\ \frac{y-d_1}{y_0-d_1}, & y \in [y_0, d_1] \end{cases} \quad (3-2)$$

The inverse function of  $A_{y_0}(y)$  that is not difficult to get is

$$A_{y_0}(y)^{-1} = \begin{cases} d_y(y_0 - c_1) + c_1, & d_y \in [0, 1] \\ d_y(y_0 - d_1) + d_1, & d_y \in [0, 1] \end{cases} \quad (3-3)$$

where  $d_y$  is the approximation-degree of  $y$  to  $y_0$ .

The graphs of the approximation-degree functions  $A_{x_0}(x)$  and  $A_{y_0}(y)$  are shown in Figure 3-1 and Figure 3-2, respectively.

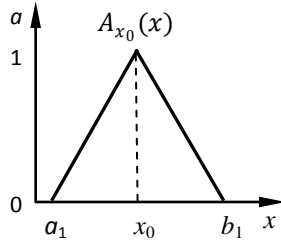


Figure 3-1 Graph of approximation-degree function  $A_{x_0}(x)$

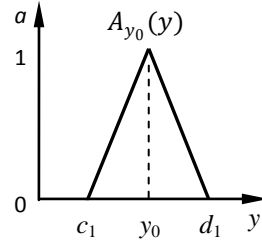


Figure 3-2 Graph of approximation-degree function  $A_{y_0}(y)$

Now we find the approximation-degree  $A_{x_0}(x')$ , and then set  $A_{y_0}(y') = A_{x_0}(x')$  (that is, transmitting the approximation-degree of  $x'$  to  $x_0$  to  $y'$ ); Further, we derive the required approximate value of  $y'$  from approximation-degree  $A_{y_0}(y')$  and inverse function  $A_{y_0}(y)^{-1}$  of approximation-degree function of  $A_{y_0}(y)$ .

It can be seen that the inverse function  $A_{y_0}(y)^{-1}$  of  $A_{y_0}(y)$  is a piecewise function, which has two parallel expressions. Thus, substituting approximation-degree  $A_{y_0}(y') = d$  into  $A_{y_0}(y)^{-1}$ , we can get two  $y$ -values ( $y_1$  and  $y_2$ ). That is to say, from one  $x'$  we obtain two  $y$ -values (as shown in Figure 3-3). Then, which  $y$ -value should be chosen as the desired approximate value of function?

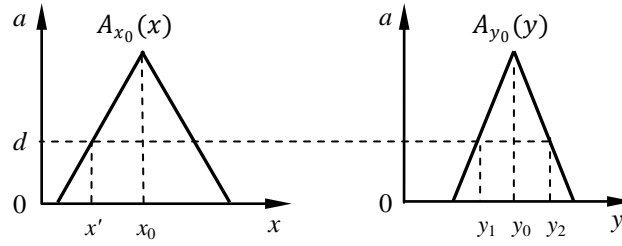


Figure 3-3 One  $x'$  is related to two  $y$ -values

Obviously, the desired  $y$ -value is related to the orientation of  $x'$  relative to  $x_0$  and the trend (i.e., being increasing, decreasing, or a constant) of  $f(x)$  near  $x_0$ . Thus, when the derivative  $f'(x_0)$  is not known, we can consider whether there is a point  $x^*$  on the  $x'$  side near the point  $x_0$ , whose corresponding value of function,  $f(x^*) = y^*$ , is known. If there is such a point  $x^*$ , we can estimate the trend of  $f(x)$  between the  $x_0$  and  $x^*$  by utilizing the size relation between the corresponding  $y^*$  and  $y_0$ , and then determine the choice of  $y$ -value. In fact, when  $x^* < x' < x_0$ , if  $y^* < y_0$ , which then shows that the general trend of function  $f(x)$  is increasing on the sub interval  $(x^*, x_0)$ , thus the  $y_1$ , i.e., that  $y$ -value less than  $y_0$ , should be chosen; while if  $y^* > y_0$ , which then shows that the general trend of function  $f(x)$  is decreasing on the subinterval  $(x^*, x_0)$ , thus the  $y_2$ , i.e., that  $y$ -value larger than  $y_0$ , should be chosen. Similarly, when  $x_0 < x' < x^*$ , if  $y_0 < y^*$ , which then shows that the general trend of function  $f(x)$  is increasing on the subinterval  $(x_0, x^*)$ , thus the  $y_2$  should be chosen; while if  $y_0 > y^*$ , which then shows that the general trend of function  $f(x)$  is decreasing on the subinterval  $(x_0, x^*)$ , thus the  $y_1$  should be chosen. But, if  $y^* = y_0$ , which then shows that the function  $f(x)$  is likely a constant on the subintervals  $(x^*, x_0)$  or  $(x_0, x^*)$ , thus  $y = y_0$  can be taken no matter  $x^* < x' < x_0$  or  $x_0 < x' < x^*$ .

### 3.2 Approximate Evaluation of Multivariate Functions Based on Approximation-Degree

Let's take the function of two variables as an example to discuss this problem.

Let  $z = f(x, y)$  be a function (relation) from  $[a_1, b_1] \times [a_2, b_2]$  to  $[c, d]$ . Suppose a pair of corresponding values,  $((x_0, y_0), z_0)$ , of function  $z = f(x, y)$  are known, and point  $(x', y')$  is approximate to point  $(x_0, y_0)$ . In the case that expression of function  $f(x, y)$  is unknown or not used, find the approximate value of  $f(x', y')$ .

This problem is similar to the problem of function of one variable above, so we extend the above treatment to function of two variables, that is, we can find the approximate value of the function at point  $(x', y')$  from the approximation-degree of point  $(x', y')$  to point  $(x_0, y_0)$ . Concretely, firstly get the approximation-degree  $A_{(x_0, y_0)}(x', y')$ , then set  $A_{z_0}(z) = A_{(x_0, y_0)}(x', y')$ ; and then, substitute it into inverse function  $A_{z_0}(z)^{-1}$  of  $A_{z_0}(z)$ . Here, there will also be two  $z$ -values. Thus, we can consider whether there is an adjacent point  $(x^*, y^*)$  of point  $(x_0, y_0)$  in the direction determined by points  $(x_0, y_0)$  and  $(x', y')$ , whose function value is known. If there exists such a point  $(x^*, y^*)$  (as shown in Figure 3-4), then we can determine the choice of  $z$ -value by taking  $f(x^*, y^*)$  as a reference.

However, although the technique is feasible in theory, it has precondition, it requires that the points  $(x^*, y^*)$ ,  $(x_0, y_0)$  and  $(x', y')$  must be exactly on a straight line. So, the method has some limitations for multivariate functions. Then, is there a better way?

By the definition of strict approximation (see Definition 2-3), point  $(x', y')$  is approximate to point  $(x_0, y_0)$  is equivalent to  $x'$  is approximate to  $x_0$  and  $y'$  is approximate to  $y_0$ . Thus, we can find the approximate values  $z_x$  and  $z_y$  of function  $f(x, y_0)$  and  $f(x_0, y)$  at  $x'$  and  $y'$ , respectively. Thus, we further imagine that if there is respectively an adjacent point of point  $(x_0, y_0)$  in the  $x$ -direction and  $y$ -direction of the point  $(x_0, y_0)$ , which is respectively  $(x^*, y_0)$  and  $(x_0, y^*)$ , as shown in Figure 3-5, and the corresponding function values  $f(x^*, y_0)$  and  $f(x_0, y^*)$  are known, then the approximate value  $z_x$  of  $f(x', y_0)$  can be got by utilizing  $f(x^*, y_0)$ , and the approximate value  $z_y$  of  $f(x_0, y')$  can be got by utilizing  $f(x_0, y^*)$ , just like that of the previous univariate function. Thus, we firstly find separately the approximation-degree  $A_{x_0}(x')$  and  $A_{y_0}(y')$ , then set  $A_{z_0}(z') = A_{x_0}(x')$ , and  $A_{z_0}(z') = A_{y_0}(y')$ ; and then, substitute them separately into inverse function  $A_{z_0}(z)^{-1}$  of  $A_{z_0}(z)$  and get two pairs of candidate  $z$ -values, then taking  $f(x^*, y_0)$  and  $f(x_0, y^*)$  as reference, choose  $z_x$  and  $z_y$  from respective candidate values, respectively (as shown in Figure 3-5).

Having got the approximate values  $z_x$  and  $z_y$ , how can we further obtain the approximate value  $z$  we need? Can  $z_x$  and  $z_y$  be synthesized into one  $z$ ? For example, take the average value or weighted sum of  $z_x$  and  $z_y$  as  $z$ . But numerical experiments show that although the average value or weighted sum can be taken as the synthesis value, but the effect is generally not very ideal. However, fortunately, on the basis of the average value, by analyzing further, we have found a better way to synthesize  $z_x$  and  $z_y$ .

Let  $z_1 = (z_x + z_y)/2$  be the average of  $z_x$  and  $z_y$ . It can be seen from Figure 3-6 that  $z_1$  can actually be viewed as an approximate value of  $f(x, y)$  at midpoint (denoted by  $(x_1, y_1)$ ) between  $(x', y_0)$  and  $(x_0, y')$ . We can see from the figure that  $z_1 < z_0$ , i.e., the varying trend of function values from  $z_0$  to  $z_1$  is decreasing. Set  $z_0 - z_1 = c_1$  ( $c_1$  is the length of segment  $BC$  in Figure 3-7(a)), then  $z_1 = z_0 - c_1$ . Then, according to the varying trend of function values from  $z_0$  to  $z_1$  (i.e., the slope of segment  $AB$  in Figure 3-7(a)), also, taking into account that point  $(x_1, y_1)$  is just the midpoint of segment joining points  $(x', y')$  and  $(x_0, y_0)$ ,

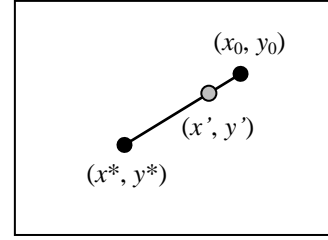


Figure 3-4 Utilizing the value of the function at point  $(x^*, y^*)$  to determine the approximate value of  $f(x', y')$

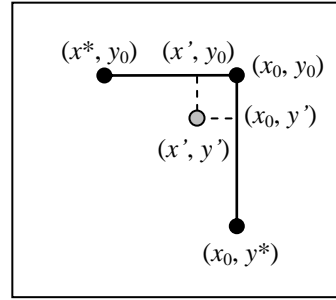


Figure 3-5 Utilizing the values of the function at points  $(x^*, y_0)$  and  $(x_0, y^*)$  to determine the approximate values of  $f(x', y_0)$  and  $f(x_0, y')$ , respectively

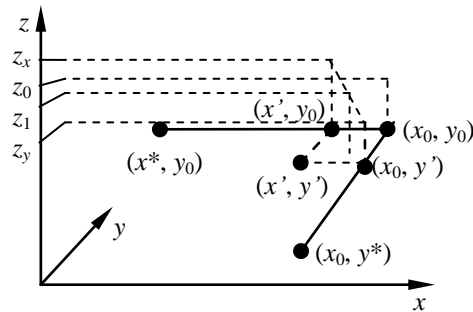


Figure 3-6 Illustration 1 of synthesizing  $z_x$  and  $z_y$  into  $z$

that is,  $\|(x', y') - (x_0, y_0)\| = 2 \cdot \|(x_1, y_1) - (x_0, y_0)\|$ , so we infer that the approximate value of function at point  $(x', y')$  can be  $z_0 - 2c_1$  (as shown in Figure 3-7(a)). Thus, it follows that

$$z = z_0 - 2c_1 = z_0 - 2(z_0 - z_1) = z_0 - 2[z_0 - (z_x + z_y)/2] = z_x + z_y - z_0$$

Of course,  $z_1$  may also be greater than  $z_0$  or equal to  $z_0$ . If  $z_1 > z_0$ , then the varying trend of function values from  $z_0$  to  $z_1$  is increasing (as shown in Figure 3-7(b)). Set  $z_0 - z_1 = c_2$  ( $c_2$  is the length of segment  $BC$  in Figure 3-7(b)), then  $z_1 = z_0 - c_2$ . Then, according to the varying trend of function values from  $z_0$  to  $z_1$  (i.e., the slope of segment  $AB$  in Figure 3-7(b)), we infer that the value of function at point  $(x', y')$  can be  $z_0 - 2c_2$ . Thus,

$$z = z_0 + 2c_2 = z_0 + 2(z_1 - z_0) = z_0 + 2[(z_x + z_y)/2 - z_0] = z_x + z_y - z_0$$

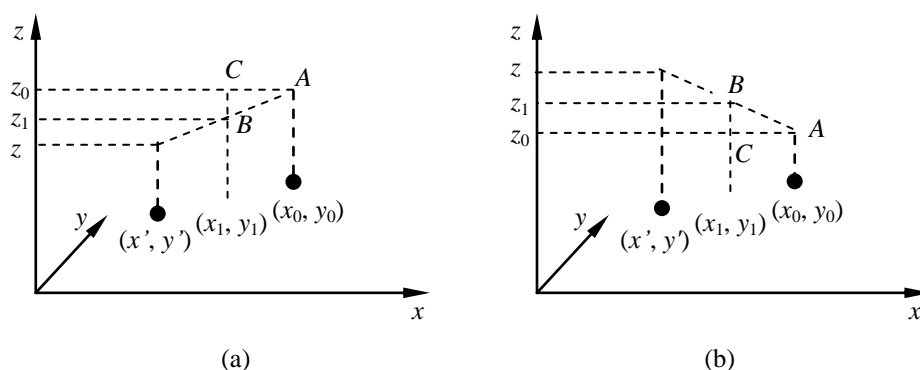


Figure 3-7 Illustration 2 of synthesizing  $z_x$  and  $z_y$  into  $z$

The third case:  $z_1 = z_0$ . This indicates that the values of function remain unchanged from  $z_0$  to  $z_1$ . Thus, we can take  $z = z_0$ . And by  $z_0 = z_1 = (z_x + z_y)/2$ , it follows that  $2z_0 = z_x + z_y$ . Thus,

$$z = 2z_0 - z_0 = z_x + z_y - z_0$$

In summary, we see that, no matter what relationship may be between the average of  $z_x$  and  $z_y$  and the  $z_0$ , or no matter how the value of the function varies from  $z_0$  to  $z_1$ , the approximate value of the function at point  $(x', y')$  can always be taken as

$$z = z_x + z_y - z_0 \quad (3-4)$$

Synthesizing the above analysis, we obtain a method for finding the approximate value of the function at the point  $(x', y')$  in the condition of that there are adjacent points  $(x^*, y_0)$  and  $(x_0, y^*)$  of point  $(x_0, y_0)$ . That is, find firstly the approximate values of  $f(x', y_0)$  and  $f(x_0, y')$ ,  $z_x$  and  $z_y$ , respectively, by  $f(x_0, y_0) = z_0$  and approximation-degrees  $A_{x_0}(x')$  and  $A_{y_0}(y')$  and by utilizing  $f(x^*, y_0)$  and  $f(x_0, y^*)$ , and then synthesize  $z_x$  and  $z_y$  into a value  $z$  by Formula (3-4) as the approximate value of  $f(x', y')$ .

Now we see that here we separate actually the approximate evaluation of a function of two variables into the approximate evaluation of two functions of one variable, then, synthesize two obtained approximate values into a value as an approximate value of the original function of two variables.

It is not difficult to see that the idea and method of “first separating then synthesizing” in the approximate evaluation of the functions of two variables above can also be extended to the approximate evaluation of the functions of three variables. That is to say, in the condition



of a pair  $((x_0, y_0, z_0), u_0)$  of corresponding values of function of three variables,  $u=f(x, y, z)$ , being known and point  $(x', y', z')$  being approximate to point  $(x_0, y_0, z_0)$ , if there is respectively an adjacent point of the point  $(x_0, y_0, z_0)$  on the sides of corresponding coordinates  $x'$ ,  $y'$  and  $z'$  of point  $(x', y', z')$  in the  $x$ -direction,  $y$ -direction and  $z$ -direction of the point  $(x_0, y_0, z_0)$ , which respectively is  $(x^*, y_0, z_0)$ ,  $(x_0, y^*, z_0)$  and  $(x_0, y_0, z^*)$  (as shown in Figure 3-8), and the corresponding values of function,  $f(x^*, y_0, z_0)$ ,  $f(x_0, y^*, z_0)$  and  $f(x_0, y_0, z^*)$ , are known, then we can find firstly the approximate values of  $f(x', y_0, z_0)$ ,  $f(x_0, y', z_0)$  and  $f(x_0, y_0, z')$ ,  $u_x$ ,  $u_y$  and  $u_z$ , respectively, by  $f(x_0, y_0, z_0)=u_0$  and approximation-degrees  $A_{x_0}(x')$ ,  $A_{y_0}(y')$  and  $A_{z_0}(z')$  and by utilizing  $f(x^*, y_0, z_0)$ ,  $f(x_0, y^*, z_0)$  and  $f(x_0, y_0, z^*)$ , and then synthesize the  $u_x$ ,  $u_y$  and  $u_z$  into a value  $u$  as the approximate value of  $f(x', y', z')$ .

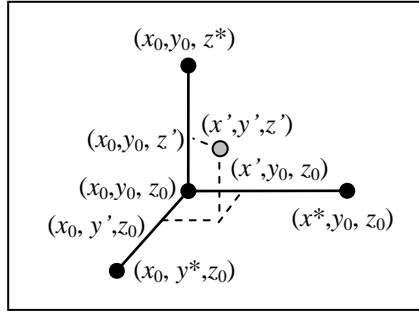


Figure 3-8 Utilizing the values of the function at points  $(x^*, y_0, z_0)$ ,  $(x_0, y^*, z_0)$  and  $(x_0, y_0, z^*)$  to determine the approximate values of  $f(x', y_0, z_0)$ ,  $f(x_0, y', z_0)$  and  $f(x_0, y_0, z')$ , respectively

From the synthesizing formula of approximate values of the function of two variables,  $z=z_x+z_y-z_0=z_x+z_y-1\bullet z_0$ , we infer that the synthesizing formula of approximate values of the function of three variables should be

$$u = u_x + u_y + u_z - 2u_0 \quad (3-5)$$

In fact, when  $(x', y', z') = (x_0, y_0, z_0)$ ,  $u_x = u_y = u_z = u_0$ , while on the other hand,  $u_0 = u_0 + u_0 + u_0 - 2u_0$ . This indicates that for the especial point  $(x_0, y_0, z_0)$ , the Formula (3-5) is correct. This indirectly verifies the rationality of the synthesizing formula (the three-dimensional AD interpolation example in Section 5.2 further confirms that the Formula (3-5) is reasonable).

Based on the above analysis, generally, in the condition of a pair  $(\mathbf{x}_0, y_0)$  ( $\mathbf{x}_0 = (x_{1_0}, x_{2_0}, \dots, x_{n_0})$ ) of corresponding values of function of  $n$  variables,  $y=f(\mathbf{x})$  ( $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ) being known and point  $\mathbf{x}' = (x_1', x_2', \dots, x_n')$  being approximate to point  $\mathbf{x}_0$ , if there is an adjacent point of the point  $\mathbf{x}_0$  on the side of corresponding coordinate of point  $\mathbf{x}'$  in the each coordinate direction of point  $\mathbf{x}_0$ , that is there exist points  $(x_1^*, x_{2_0}, \dots, x_{n_0})$ ,  $(x_{1_0}, x_2^*, \dots, x_{n_0})$ , ..., and  $(x_{1_0}, x_{2_0}, \dots, x_n^*)$ , and the corresponding values of function are known, then we can find firstly the approximate values of  $f(x_1', x_{2_0}, \dots, x_{n_0})$ ,  $f(x_{1_0}, x_2', \dots, x_{n_0})$ , ..., and  $f(x_{1_0}, x_{2_0}, \dots, x_n')$ ,  $y_{x_1}, y_{x_2}, \dots, y_{x_n}$ , respectively, by  $f(\mathbf{x}_0) = y_0$  and approximation-degrees  $A_{x_0}(x_1')$ ,  $A_{x_0}(x_2')$ , ...,  $A_{x_0}(x_n')$  and by utilizing the values of function at these adjacent points,  $f(x_1^*, x_{2_0}, \dots, x_{n_0})$ ,  $f(x_{1_0}, x_2^*, \dots, x_{n_0})$ , ...,  $f(x_{1_0}, x_{2_0}, \dots, x_n^*)$ , and then synthesize the  $y_{x_1}, y_{x_2}, \dots, y_{x_n}$  into a value  $y$  as the approximate value of  $f(\mathbf{x}')$ , and the formula of synthesizing these approximate values is



$$y = \sum_{i=1}^n y_{x_i} - (n-1)y_0 \quad (3-6)$$

In this way, we obtain an effective approach and method to find approximate values of multivariate functions.

For convenience of narration, we may as well refer to the formula (3-6) as the **Sum-Times-Difference formula**.

#### 4 Interpolation Based on Approximation-Degree

Let  $y=f(x)$  be a function (relation) from  $[a, b]$  to  $[c, d]$ , and  $\{(x_i, y_i)\}_{i=1}^n$  be a set of pairs of corresponding values of function  $y=f(x)$ . Require to construct an interpolating function  $g(x)$  in the case that the expression of function  $f(x)$  is unknown or not used, such that  $g(x_i)=f(x_i)$  ( $i=1, 2, \dots, n$ ), and for other  $x \in [a, b]$ ,  $g(x) \approx f(x)$ . This is the usual interpolation problem.

The readers may have noticed that the method finding functional approximate value based on approximation-degree above can also be used for interpolation. Below we discuss specifically this problem.

Let  $a=x_1 < x_2 < \dots < x_n=b$ , then  $\{x_i\}_{i=1}^n$  is a set of interpolation base points (or nodes). We define the approximation domain of  $x_1$  is  $[x_1, x_2]$ , the approximation domains of  $x_i$  are  $[x_{i-1}, x_{i+1}]$  ( $i=2, 3, \dots, n-1$ ), and the approximation domain of  $x_n$  is  $[x_{n-1}, x_n]$ , and define the approximation-degree function of base point  $x_1$  is

$$A_{x_1}(x) = \frac{x-x_2}{x_1-x_2}, \quad x \in [x_1, x_2] \quad (4-1)$$

The approximation-degree functions of  $x_i$  is

$$A_{x_i}(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x-x_{i+1}}{x_i-x_{i+1}}, & x \in [x_i, x_{i+1}] \end{cases} \quad (4-2)$$

And the approximation-degree function of  $x_n$  is

$$A_{x_n}(x) = \frac{x-x_{n-1}}{x_n-x_{n-1}}, \quad x \in [x_{n-1}, x_n] \quad (4-3)$$

The graphs of these functions are shown in Figure 4-1.

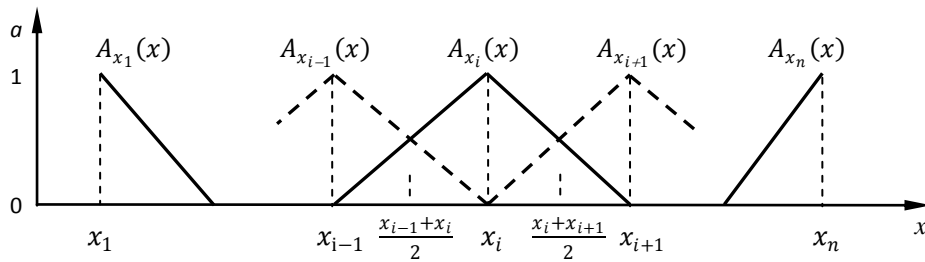


Figure 4-1 Graphs of the approximation-degree functions of numbers  $x_1, x_1, \dots, x_n$

### Notes:

• Here, the approximation domain of a number  $x_i$  is determined not only by  $x_i$  as before, but by two neighbors of  $x_i$ ,  $x_{i-1}$  and  $x_{i+1}$ . Therefore, two approximation radii  $rl = x_i - x_{i-1}$  and  $rr = x_{i+1} - x_i$  of the number  $x_i$  defined by Equation (4-2) are not necessarily equal. For the sake of distinction, we call the approximation domain determined by a number alone to be the **absolute approximation domain** of the number, and the approximation domain determined by its two neighbors to be the **relative approximation domain** of the number. In the absence of a special statement, the subsequent approximation domains refer to the relative approximation domains.

• When  $x \in [x_{i-1}, \frac{x_{i-1}+x_i}{2})$ ,  $A_{x_i}(x) < 0.5$  and  $A_{x_{i-1}}(x) > 0.5$ , this shows that  $x$  is closer to  $x_{i-1}$ ; similarly, when  $x \in (\frac{x_i+x_{i+1}}{2}, x_{i+1}]$ ,  $A_{x_i}(x) < 0.5$  and  $A_{x_{i+1}}(x) > 0.5$ , this shows that  $x$  is closer to  $x_{i+1}$ ; and only when  $x \in [\frac{x_i+x_{i-1}}{2}, \frac{x_i+x_{i+1}}{2}]$ ,  $A_{x_i}(x) \geq 0.5$ . If we make an appointment: a point  $x$  is practically approximate to the base point  $x_i$ , if and only if its approximation-degree  $A_{x_i}(x) \geq 0.5$ , then interval  $[\frac{x_i+x_{i-1}}{2}, \frac{x_i+x_{i+1}}{2}]$  is the **practical domain** of approximation-degree function  $A_{x_i}(x)$ , and that is also, the **practical approximation domain** of  $x_i$  (while interval  $[x_{i-1}, x_{i+1}]$  is the **conceptual domain** of  $A_{x_i}(x)$  and **conceptual approximation domain** of  $x_i$ ). And then, in the sense of practical approximation, we have the following conclusion.

**Proposition 4-1** A point  $x$  is practically approximate to the base point  $x_i$  (i.e.  $A_{x_i}(x) \geq 0.5$ ), if and only if  $x$  lies in the practical approximation domain  $([\frac{x_i+x_{i-1}}{2}, \frac{x_i+x_{i+1}}{2}])$  of the  $x_i$ .

We then define the approximation-degree functions of  $y_i$  ( $i=1, 2, \dots, n$ ) in the same principle and way.

First, we point out that  $y_1, y_2, \dots, y_n$  corresponding to  $x_1, x_2, \dots, x_n$  in order do not necessarily satisfy  $y_1 < y_2 < \dots < y_n$ . In fact, the size relationship between  $y_i$  and  $y_{i-1}$  in the sample data may be  $y_i < y_{i-1}$ ,  $y_i = y_{i-1}$ , or  $y_i > y_{i-1}$ ; the size relationship between  $y_i$  and  $y_{i+1}$  in the sample data may be  $y_i < y_{i+1}$ ,  $y_i = y_{i+1}$ , or  $y_i > y_{i+1}$ . We also noticed that when  $x$  lies between  $x_{i-1}$  and  $x_i$ , the corresponding  $y$  lies certainly between  $y_{i-1}$  and  $y_i$ ; and when  $x$  lies between  $x_i$  and  $x_{i+1}$ , the corresponding  $y$  lies certainly between  $y_i$  and  $y_{i+1}$ . Thus, we only need to consider separately four (semi) approximation-degree functions of number  $y_i$  in the four cases of  $y_{i-1} < y_i$ ,  $y_i < y_{i-1}$ ,  $y_i < y_{i+1}$ , and  $y_{i+1} < y_i$ .

(1) Suppose  $y_{i-1} < y_i$ . We define the first sub expression of the approximation-degree function of  $y_i$  in this case to be

$$A_{y_i}(y) = \frac{y - y_{i-1}}{y_i - y_{i-1}} = \frac{1}{y_i - y_{i-1}} y - \frac{y_{i-1}}{y_i - y_{i-1}}, \quad y \in [y_{i-1}, y_i] \quad (4-4)$$

Its graph is shown in Figure 4-2(a).

(2) Suppose  $y_i < y_{i-1}$ . We define the second sub expression of the approximation-degree function of  $y_i$  in this case to be

$$A_{y_i}(y) = \frac{y_{i-1} - y}{y_{i-1} - y_i} = \frac{1}{y_i - y_{i-1}} y - \frac{y_{i-1}}{y_i - y_{i-1}}, \quad y \in [y_i, y_{i-1}] \quad (4-5)$$

Its graph is shown in Figure 4-2(b).

(3) Suppose  $y_i < y_{i+1}$ . We define the second sub expression of the approximation-degree function of  $y_i$  in this case to be

$$A_{y_i}(y) = \frac{y_{i+1}-y}{y_{i+1}-y_i} = \frac{1}{y_i-y_{i+1}} y - \frac{y_{i+1}}{y_i-y_{i+1}}, \quad y \in [y_i, y_{i+1}] \quad (4-6)$$

Its graph is shown in Figure 4-2(c).

(4) Suppose  $y_{i+1} < y_i$ . We define the first sub expression of the approximation-degree function of  $y_i$  in this case to be

$$A_{y_i}(y) = \frac{y-y_{i+1}}{y_i-y_{i+1}} = \frac{1}{y_i-y_{i+1}} y - \frac{y_{i+1}}{y_i-y_{i+1}}, \quad y \in [y_{i+1}, y_i] \quad (4-7)$$

Its graph is shown in Figure 4-2(d).

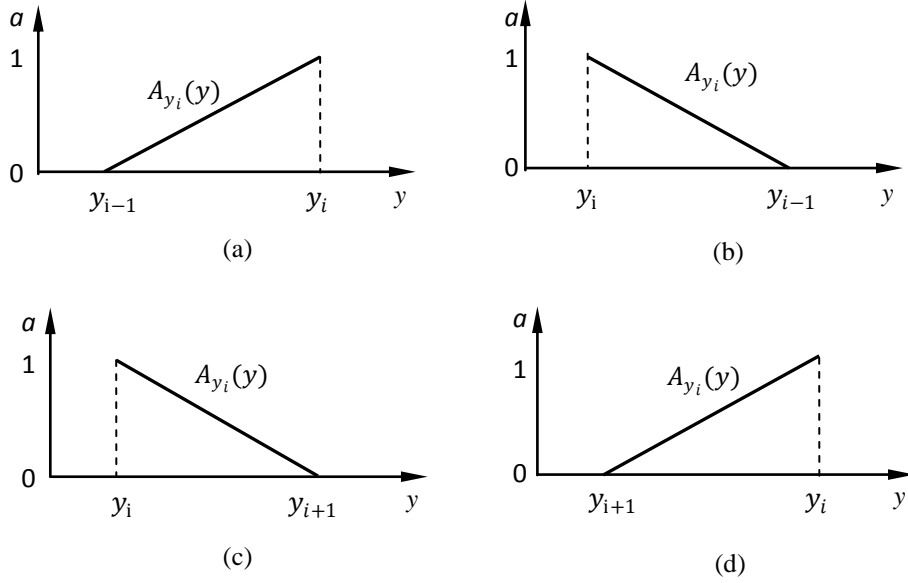


Figure 4-2 Approximation-degree functions of number  $y_i$  under different conditions

Obviously, in the four expressions above, (4-4)=(4-5) and (4-6)=(4-7). That is to say, no matter  $y_{i-1} < y_i$  or  $y_i < y_{i-1}$ , when  $y$  lies between  $y_{i-1}$  and  $y_i$ , the expression of approximation-degree function of  $y_i$  is always

$$A_{y_i}(y) = \frac{y-y_{i-1}}{y_i-y_{i-1}} = \frac{1}{y_i-y_{i-1}} y - \frac{y_{i-1}}{y_i-y_{i-1}}$$

and no matter  $y_{i+1} < y_i$  or  $y_i < y_{i+1}$ , when  $y$  lies between  $y_i$  and  $y_{i+1}$ , the expression of approximation-degree function of  $y_i$  is always

$$A_{y_i}(y) = \frac{y-y_{i+1}}{y_i-y_{i+1}} = \frac{1}{y_i-y_{i+1}} y - \frac{y_{i+1}}{y_i-y_{i+1}}$$

Thus, the above 4 functional expressions can be reduced as two expressions.

Further, it can be seen that the coefficients of the first degree terms,  $\frac{1}{y_i-y_{i-1}}$  and  $\frac{1}{y_i-y_{i+1}}$ , in these two expressions are just the slopes of the corresponding straight lines. So they being positive or negative decide the corresponding function to be increasing or decreasing. Thus, function  $y=f(x)$  being incremented or decremented on the subinterval  $[\frac{x_i+x_{i-1}}{2}, x_i]$  and

$[x_i, \frac{x_i+x_{i+1}}{2}]$  are respectively determined by the signs of differences  $y_i-y_{i-1}$  and  $y_i-y_{i+1}$ . This means that the numerical calculation takes the place of the logical judgment for the trend of function  $f(x)$  on the subintervals  $[\frac{x_i+x_{i-1}}{2}, x_i]$  and  $[x_i, \frac{x_i+x_{i+1}}{2}]$ .

And then, we get the inverse expressions of these two functional expressions, respectively

$$A_{y_i}(y)^{-1} = d_y(y_i - y_{i-1}) + y_{i-1} \quad (4-8)$$

and

$$A_{y_i}(y)^{-1} = d_y(y_i - y_{i+1}) + y_{i+1} \quad (4-9)$$

Here  $d_y = A_{y_i}(y) \in [0, 1]$  is the approximation-degree of  $y$  to  $y_i$ .

Now, we transmit the approximation-degree of  $x$  ( $x \in [\frac{x_i+x_{i-1}}{2}, \frac{x_i+x_{i+1}}{2}]$ ) to  $x_i$  to the corresponding  $y$ , that is, set

$$d_y = d_x = A_{x_i}(x)$$

Also, considering that on the subinterval  $[\frac{x_i+x_{i-1}}{2}, x_i]$ , the interpolated function  $y=f(x)$  may being increasing, decreasing, or a constant; while when  $y=f(x)$  is increasing, certainly  $y_{i-1} < y_i$ , so the corresponding  $y \in [\frac{y_i+y_{i-1}}{2}, y_i]$ ; when  $y=f(x)$  is decreasing, certainly  $y_i < y_{i-1}$ , so the corresponding  $y \in [y_i, \frac{y_i+y_{i-1}}{2}]$ ; and when  $y=f(x)$  is a constant,  $y_i = y_{i-1}$ , so the corresponding  $y = y_i \in [\frac{y_i+y_{i-1}}{2}, y_i]$  as well as  $y \in [y_i, \frac{y_i+y_{i-1}}{2}]$ . Thus, when  $x \in [\frac{x_i+x_{i-1}}{2}, x_i]$ , the corresponding  $y_i$  and  $y_{i-1}$  are adjacent, and here  $A_{x_i}(x) = \frac{x-x_{i-1}}{x_i-x_{i-1}}$ , thus, the expression of the corresponding inverse function  $A_{y_i}(y)^{-1}$  in Equation (4-8) becomes

$$\begin{aligned} A_{y_i}(y)^{-1} &= \frac{x-x_{i-1}}{x_i-x_{i-1}}(y_i - y_{i-1}) + y_{i-1} \\ &= \frac{y_i - y_{i-1}}{x_i - x_{i-1}}x + \frac{x_i y_{i-1} - x_{i-1} y_i}{x_i - x_{i-1}} \end{aligned}$$

namely

$$y = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}x + \frac{x_i y_{i-1} - x_{i-1} y_i}{x_i - x_{i-1}}, \quad x \in [\frac{x_i+x_{i-1}}{2}, x_i] \quad (4-10)$$

Similarly, on the subinterval  $[x_i, \frac{x_i+x_{i+1}}{2}]$ , the function  $y=f(x)$  may being increasing, decreasing, or a constant; while when  $y=f(x)$  is increasing, certainly  $y_i < y_{i+1}$ , so the corresponding  $y \in [y_i, \frac{y_i+y_{i+1}}{2}]$ ; when  $y=f(x)$  is decreasing, certainly  $y_{i+1} < y_i$ , so the corresponding  $y \in [\frac{y_i+y_{i+1}}{2}, y_i]$ ; and when  $y=f(x)$  is a constant,  $y_i = y_{i+1}$ , so the corresponding  $y = y_i \in [\frac{y_i+y_{i+1}}{2}, y_i]$  as well as  $y \in [y_i, \frac{y_i+y_{i+1}}{2}]$ . Thus, when  $x \in [x_i, \frac{x_i+x_{i+1}}{2}]$ , the corresponding  $y_i$

and  $y_{i+1}$  are adjacent, and here  $A_{x_i}(x) = \frac{x_{i+1}-x}{x_{i+1}-x_i}$ , thus, the expression of the corresponding inverse function  $A_{y_i}(y)^{-1}$  in Equation (4-9) becomes

$$A_{y_i}(y)^{-1} = \frac{x_{i+1}-x}{x_{i+1}-x_i}(y_i - y_{i+1}) + y_{i+1} \\ = \frac{y_i - y_{i+1}}{x_i - x_{i+1}}x + \frac{x_i y_{i+1} - x_{i+1} y_i}{x_i - x_{i+1}}$$

namely

$$y = \frac{y_i - y_{i+1}}{x_i - x_{i+1}}x + \frac{x_i y_{i+1} - x_{i+1} y_i}{x_i - x_{i+1}}, \quad x \in [x_i, \frac{x_i + x_{i+1}}{2}] \quad (4-11)$$

Thus, with the two expressions, we can obtain directly the corresponding  $y \in [\frac{y_i + y_{i-1}}{2}, y_i]$

or  $[y_i, \frac{y_i + y_{i+1}}{2}]$  from  $x \in [\frac{x_i + x_{i-1}}{2}, x_i]$ , and obtain directly the corresponding  $y \in [y_i, \frac{y_i + y_{i+1}}{2}]$

or  $[\frac{y_i + y_{i+1}}{2}, y_i]$  from  $x \in [x_i, \frac{x_i + x_{i+1}}{2}]$ , but do not need to get approximation-degree and to evaluate inverse function.

Actually, Equations (4-10) and (4-11) are two interpolating formulas. Thus, using these two interpolation formulas, we can solve the interpolation problem mentioned above. In fact, for the evaluated point  $x \in [a, b]$ , we only need to seek firstly the approximation domain the point  $x$  lies (This is equivalent to seeking nearest base point  $x_i$  ( $i \in \{1, 2, \dots, n\}$ )), then to construct and use the corresponding interpolating formula according to the position of  $x$  relative to  $x_i$  to do interpolation.

In this way, we actually derive an interpolation method by the approximate evaluation of function based on approximation-degree. We call this method to be the **approximation-degree-based interpolation**, or **AD interpolation** for short.

Specifically, the practice of AD interpolation is: take base points (or, nodes)  $a = x_1, x_2, \dots, x_n = b$  as points of view, according to base points and their approximation domains to partition interval  $[a, b] = [x_1, x_n]$  into  $2n-2$  subintervals as shown in Figure 4-3,  $[x_1, \frac{x_1 + x_2}{2}]$ ,

$[\frac{x_1 + x_2}{2}, x_2], [x_2, \frac{x_2 + x_3}{2}], \dots, [\frac{x_{n-1} + x_n}{2}, x_n]$ , as interpolation intervals; then, for evaluated point

$x \in [\frac{x_i + x_{i-1}}{2}, x_i]$  do interpolating with Formula (4-10), for evaluated point  $x \in [x_i, \frac{x_i + x_{i+1}}{2}]$  do

interpolating with Formula (4-11). Since each specific interpolating formula implies the trend of interpolated function  $y=f(x)$  on the corresponding subintervals, therefore, there is no much error between the obtained approximate value and the expected value, and there would not occur the case that two  $y$ -values are got from an  $x$ .

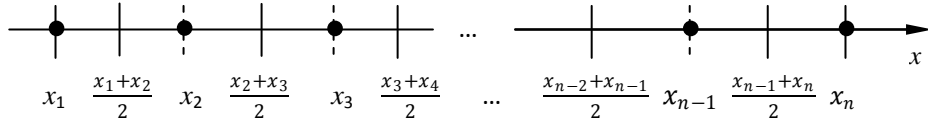


Figure 4-3 Illustration of interpolation intervals partitioned by base points and their approximation domains in one-dimensional interpolation

Where the solid lines indicate the boundaries between approximation domains of base points  $x_i$ , and a dotted line further divides an approximation domain into two parts

**Example 4.1** Use AD interpolation to do interpolating to functions  $y=\sin x$  and  $y=\cos x$ , respectively.

We take sampled-data points of  $y=\sin x$ ,  $(x_i, y_i)$ , as follows:

$x_i$ :  $0 \times \pi, 0.1 \times \pi, 0.2 \times \pi, 0.3 \times \pi, \dots, 1.9 \times \pi, 2 \times \pi$ .

$y_i$ :  $\sin(x_i)$ .

and take evaluated points  $x$ :  $0 \times \pi, 0.02 \times \pi, 0.04 \times \pi, 0.06 \times \pi, \dots, 1.98 \times \pi, 2 \times \pi$ .

By using our AD interpolation to do interpolating (programmed with MATLAB), the corresponding  $y$ -values obtained are as follows:

0	0.0624	0.1249	0.1873	0.2497	0.3118	0.3681	0.4245	0.4808
0.5371	0.5923	0.6369	0.6816	0.7263	0.7710	0.8133	0.8420	0.8707
0.8994	0.9281	0.9530	0.9629	0.9728	0.9827	0.9926	0.9975	0.9876
0.9778	0.9679	0.9580	0.9424	0.9138	0.8851	0.8564	0.8277	0.7934
0.7487	0.7040	0.6593	0.6146	0.5653	0.5089	0.4526	0.3963	0.3400
0.2809	0.2185	0.1561	0.0936	0.0312	-0.0312	-0.0936	-0.1561	-0.2185
-0.2809	-0.3400	-0.3963	-0.4526	-0.5089	-0.5653	-0.6146	-0.6593	
-0.7040	-0.7487	-0.7934	-0.8277	-0.8564	-0.8851	-0.9138	-0.9424	-0.9580
-0.9679	-0.9778	-0.9876	-0.9975	-0.9926	-0.9827	-0.9728	-0.9629	
-0.9530	-0.9281	-0.8994	-0.8707	-0.8420	-0.8133	-0.7710	-0.7263	-0.6816
-0.6369	-0.5923	-0.5371	-0.4808	-0.4245	-0.3681	-0.3118	-0.2497	-0.1873
-0.1249	-0.0624	0.0000						

The interpolation effect is shown in Figure 4-4(a).

To function  $y=\cos x$ , we do interpolating with the same method and data, the interpolation effect is shown in Figure 4-4(b).

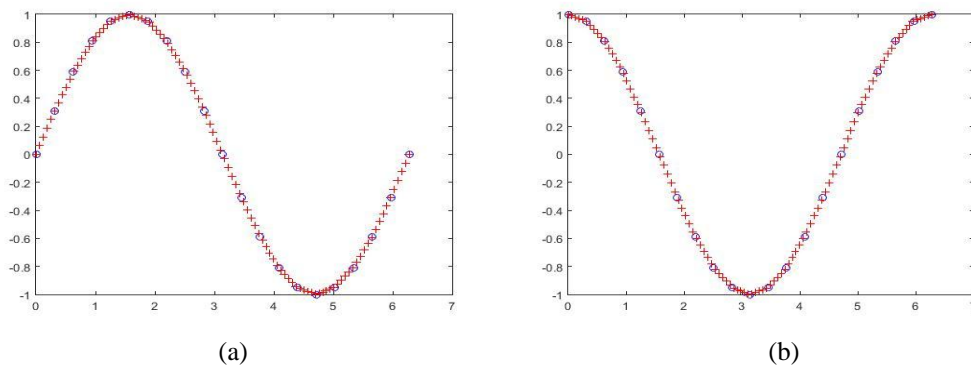


Figure 4-4 The effect drawings of AD interpolation to functions  $y=\sin x$  and  $y=\cos x$

Where the blue circles, “o”, in the graph indicate the sample data points, and the red crosses, “+”, indicate the points obtained by AD interpolation.

### **Accuracy analysis:**

sin  $x$ : maximum error 0.0121, minimum error 0, average error 0.0052

cos  $x$ : maximum error 0.0122, minimum error 0, average error 0.0052

Actually, it can be seen from the interpolation formulas that one-dimensional AD interpolation is a local linear interpolation. That is to say that this one-dimensional AD interpolation has the effect of “reaching the same goal from different routes” with usual local linear interpolation.

## **5 Multidimensional and High-Dimensional Interpolations Based on Approximation-Degree**

### **5.1 Two-Dimensional AD Interpolation**

Suppose some data points of function of two variables,  $z=f(x, y)$ ,  $((x_i, y_j), z_{ij})$  ( $i=1,2,\dots,n$ ,  $j=1,2,\dots,m$ ), are known, and the base points  $(x_i, y_j)$  are regularly distributed, that is, they satisfy  $a_x=x_1<x_2<\dots,<x_n=b_x$  and  $a_y=y_1<y_2<\dots,<y_m=b_y$ , and can also form an  $m \times n$  matrix:

$$\begin{pmatrix} (x_1, y_1), (x_2, y_1), \dots, (x_n, y_1) \\ (x_1, y_2), (x_2, y_2), \dots, (x_n, y_2) \\ \dots & \dots & \dots \\ (x_1, y_m), (x_2, y_m), \dots, (x_n, y_m) \end{pmatrix}$$

Correspondingly, they form a Cartesian grid on the  $x$ - $y$  plane (as shown in Figure 5-1). We consider corresponding interpolation problem.

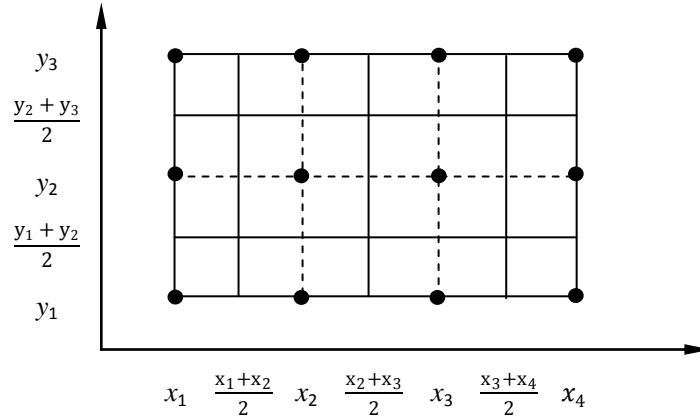


Figure 5-1 Illustration of interpolation regions partitioned by base points and their approximation domains in two-dimensional AD interpolation

Where the regions enclosed by solid lines are the approximation domains of base points  $(x_i, y_j)$ , and the dotted lines further divide an approximation domain into two or four parts

For this kind of two-dimensional interpolation (actually, this kind of interpolation is also usual two-dimensional interpolation), we use firstly the idea and technique similar to AD interpolation in Section 4 above, that is, take base points  $(x_1, y_1), \dots, (x_n, y_m)$  as points of view, according to base points and their approximation domains to partition region  $U=[a_x, b_x] \times [a_y, b_y]=[x_1, x_n] \times [y_1, y_m]$  into  $(2n-2) \times (2m-2)$  sub-regions as shown in Figure 5-1 as



interpolation regions; then, for evaluated point  $(x, y) \in U$ , seek the sub-region it lies (This is equivalent to seek the corresponding nearest base point  $(x_i, y_j)$  ( $i \in \{1, 2, \dots, n\}$ ,  $j \in \{1, 2, \dots, m\}$ ) and determine the position of  $(x, y)$  relative to  $(x_i, y_j)$ ), and then to do interpolation.

However, for the interpolation on these two-dimensional sub-regions, we have not derived the corresponding interpolating formulas. Since these base points are regularly distributed, that is, each base point has the adjacent base point(s) in the direction of its coordinates, so we use the technique of “first separating then synthesizing” described in Section 3.2 above to get the interpolated value at point  $(x, y)$ . Specifically, first separate interpolating computation of evaluated point  $(x, y)$  with corresponding nearest base point  $(x_i, y_j)$  into interpolating computation of  $x$  with  $x_i$  and interpolating computation of  $y$  with  $y_j$ , and getting respectively the corresponding approximate values  $z_x$  and  $z_y$ ; then synthesize  $z_x$  and  $z_y$  into a value  $z$  as the interpolated value at point  $(x, y)$ .

Obviously, the two interpolating computations obtained from the separation are already one-dimensional interpolating computations, so previous two interpolating formulas can be used. But, here the corresponding interpolating formulas are with respect to  $x$  and  $y$ , respectively, and which have the total of four formulas separated as two pairs:

$$\left\{ \begin{aligned} z &= \frac{z_{ij} - z_{i-1,j}}{x_i - x_{i-1}} x + \frac{x_i z_{i-1,j} - x_{i-1} z_{ij}}{x_i - x_{i-1}}, & x \in \left[ \frac{x_i + x_{i-1}}{2}, x_i \right] \end{aligned} \right. \quad (5-1)$$

$$\left\{ \begin{aligned} z &= \frac{z_{ij} - z_{i+1,j}}{x_i - x_{i+1}} x + \frac{x_i z_{i+1,j} - x_{i+1} z_{ij}}{x_i - x_{i+1}}, & x \in \left[ x_i, \frac{x_i + x_{i+1}}{2} \right] \end{aligned} \right. \quad (5-2)$$

$$\left\{ \begin{aligned} z &= \frac{z_{ij} - z_{i,j-1}}{y_j - y_{j-1}} y + \frac{y_j z_{i,j-1} - y_{j-1} z_{ij}}{y_j - y_{j-1}}, & y \in \left[ \frac{y_j + y_{j-1}}{2}, y_j \right] \end{aligned} \right. \quad (5-3)$$

$$\left\{ \begin{aligned} z &= \frac{z_{ij} - z_{i,j+1}}{y_j - y_{j+1}} y + \frac{y_j z_{i,j+1} - y_{j+1} z_{ij}}{y_j - y_{j+1}}, & y \in \left[ y_j, \frac{y_j + y_{j+1}}{2} \right] \end{aligned} \right. \quad (5-4)$$

where  $z_{ij} = f(x_i, y_j)$ .

Of course, for an evaluated point  $(x, y) \in U$ , we need only choose a formula with respect to  $x$  and a formula with respect to  $y$ . As to which two formulas will be used, it should be determined according to the subintervals that  $x$  and  $y$  lie respectively. For example, if  $x \in \left[ \frac{x_i + x_{i-1}}{2}, x_i \right]$  and  $y \in \left[ y_j, \frac{y_j + y_{j+1}}{2} \right]$ , then Formulas (5-1) and (5-4) should be respectively used.

After getting corresponding  $z_x$  and  $z_y$  by two formulas used, it is must to synthesize them into a value  $z$ . From Formula (3-7) in Section 3.2, here the formula that synthesizes  $z_x$  and  $z_y$  should be

$$z = z_x + z_y - z_{ij} \quad (5-5)$$

In this way, we have extended the AD interpolation to two-dimensional interpolation.

From the Formula (5-5) it can be seen that this kind of two-dimensional AD interpolation of “first separating then synthesizing” is different from the usual bilinear interpolation, but its effect is comparable to that of the latter.

**Example 5.1** Use AD interpolation to do interpolating to function  $z = -x^2 - y^2$ .

We take sampled-data points of  $z = -x^2 - y^2$ ,  $((x_i, y_j), z_{ij})$ , as follows:

$$x_i: -20, -18, -16, \dots, -2, 0, 2, \dots, 16, 18, 20.$$

$$y_j: -20, -18, -16, \dots, -2, 0, 2, \dots, 16, 18, 20.$$

$$z_{ij}: -x_i^2 - y_j^2$$

and take evaluated points  $(x, y)$  as follows:

$x$ : -20, -20, 20, -19.5, -17.8, -18, -15.3, -12, -10.2, -10, -10, 0, 0, 10, 10, 5.6, 4.7, -3.4, -1.8, -2.3, -3.6, 1.2, -5.4, -15.6, -20, -20, -20, -18.3, 18.4, 17.5, 16.2, 14.5, 11.1, -5.4, -12.1, -8.5, -13.9, -7.5, -7.8, -9.8, -12.4, -13.5, -14.6, -17.5, -17.8.

$y$ : -20, 20, -20, -19.5, -17.8, -5, -15.5, 2.5, -10.2, 10, -20, 0, -20, -20, -10, -15.3, -3.8, -13.4, -2.8, -1.9, -5.6, -10.2, -6.5, 5.6, 0, -10, 10, 10.4, -18.1, -16.3, -14.4, -12.3, -6.3, -15.8, -8.2, -15.6, 0.9, 1.6, 3.2, 4.6, 6.6, 2.8, -0.9, 18.6, 13.2.

Doing interpolation by using AD interpolation (programming with MATLAB), the corresponding  $z$ -values obtained are as follows:

-800.0000	-800.0000	-800.0000	-762.0000	-634.4000	-350.0000	-476.0000
-151.0000	-208.8000	-200.0000	-500.0000	0	-400.0000	-500.0000
-200.0000	-267.0000	-37.8000	-192.8000	-12.4000	-9.6000	-45.6000
-106.8000	-73.0000	-276.0000	-400.0000	-500.0000	-500.0000	-444.2000
-667.0000	-573.2000	-470.8000	-362.8000	-164.4000	-280.0000	-214.2000
-317.0000	-195.2000	-60.2000	-72.4000	-118.4000	-198.8000	-191.8000
-215.8000	-653.8000	-492.4000				

The interpolation effect is shown in Figure 5-2.

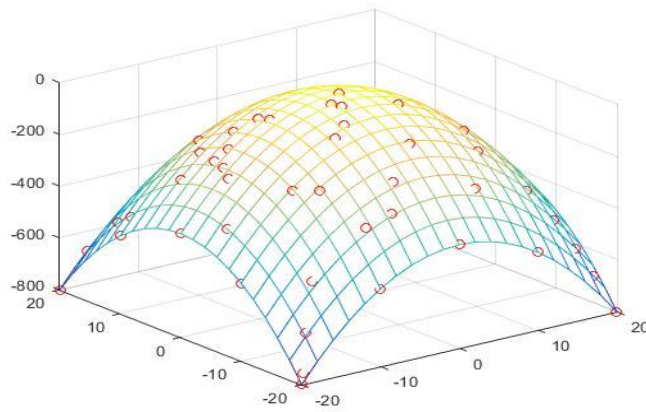


Figure 5-2 The effect drawing of scattered AD interpolation to function  $z = -x^2 - y^2$

Where the grid curve is the graph of function  $z = -x^2 - y^2$ , and the red circles indicate the points obtained by AD interpolation.

#### Accuracy analysis:

maximum error 1.8300, minimum error 0, average error 0.9222

**Example 5.2** Use AD interpolation to do interpolating to function  $z = \frac{1}{4}x^2 - \frac{1}{4}y^2$ .

We take sampled-data points,  $((x_i, y_j), z_{ij})$ , as follows:

$x_i$ : -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5.

$y_j$ : -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5.

$$z_{ij}: \frac{1}{4}x_i^2 - \frac{1}{4}y_j^2.$$

and take evaluated points  $(x, y)$  as follows:

$x = -5, -4.9, -4.8, -4.7, -4.6, -4.5, \dots, 4.5, 4.6, 4.7, 4.8, 4.9, 5.$

$y = x.$

Doing interpolation by using AD interpolation (programming with MATLAB, the corresponding  $z$ -values obtained are omitted), the interpolation effect is shown in Figure 5-3.

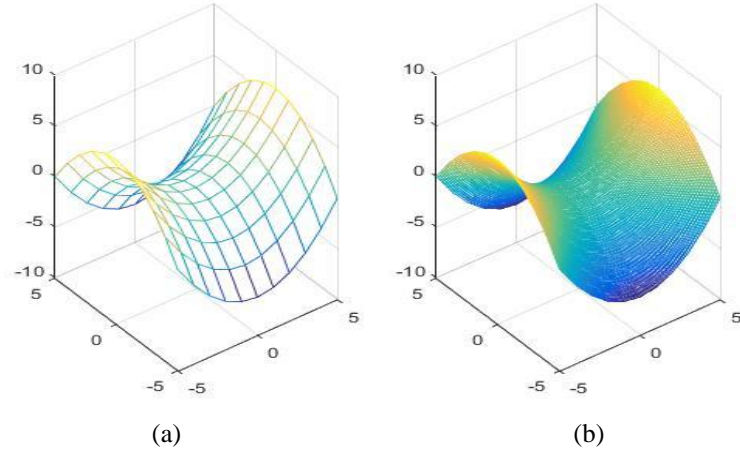


Figure 5-3 The effect drawing of AD interpolation to function  $z = \frac{1}{4}x^2 - \frac{1}{4}y^2$

Where the (a) is the functional graph before interpolating, and the (b) is the functional graph after interpolating

#### Accuracy analysis:

maximum error 0.0625, minimum error 0, average error 0.0216

**Example 5.3** Figure 5-4 below shows also an effect drawing of AD interpolation, the sampled-data (are omitted) are taken from the classic peaks function in MATLAB.

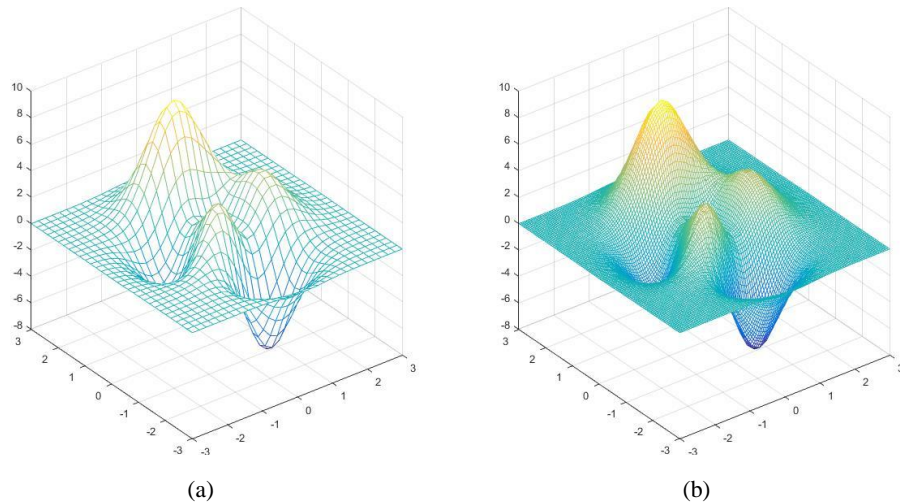


Figure 5-4 An illustration of the effect of AD interpolation

Where (a) is the functional graph formed by sample data, and (b) is the functional graph obtained by AD interpolation.

#### Accuracy analysis:

maximum error 0.3177, minimum error 0, average error 0.0224

## 5.2 Three-Dimensional AD Interpolation

Suppose some data points of function of three variables,  $u=f(x, y, z)$ ,  $((x_i, y_j, z_k), u_{ijk})$ ,  $(i=\{1,2,\dots,n\}, j=\{1,2,\dots,m\}, k=\{1,2,\dots,l\})$  are known, and the base points  $(x_i, y_j, z_k)$  are regularly distributed, that is, they satisfy  $a_x=x_1<x_2<\dots<x_n=b_x$ ,  $a_y=y_1<y_2<\dots<y_m=b_y$  and  $a_z=z_1<z_2<\dots<z_l=b_z$ , and can also form a  $m \times n \times l$  “three-dimensional matrix” (as shown in Figure 5-5), correspondingly, they form a Cartesian grid in the  $x$ - $y$ - $z$  space (as shown in Figure 5-6). We consider the corresponding interpolation problem.

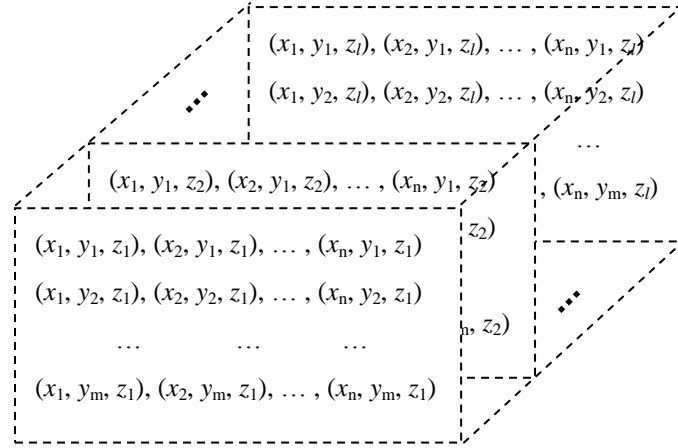


Figure5-5 A “three-dimensional matrix” formed by known base points,  $(x_i, y_j, z_k)$

For the three-dimensional interpolation, we use still the technique similar to the two-dimensional interpolation above. Specifically speaking, it is that take base points  $(x_1, y_1, z_1), \dots, (x_n, y_m, z_l)$  as points of view, according to these base points and their approximation domains to partition region  $U=[a_x, b_x] \times [a_y, b_y] \times [a_z, b_z]$  into  $(2n-2) \times (2m-2) \times (2l-2)$  sub regions as shown in Figure 5-6 as interpolation regions; then separate the interpolating computation of evaluated point  $(x, y, z)$  with corresponding nearest base point  $(x_i, y_j, z_k)$  ( $i \in \{1,2,\dots,n\}, j \in \{1,2,\dots,m\}, k \in \{1,2,\dots,l\}$ ) into the interpolating computation of  $x$  with  $x_i$ , the interpolating computation of  $y$  with  $y_j$ , and the interpolating computation of  $z$  with  $z_k$ , and get the corresponding approximate values  $u_x, u_y$ , and  $u_z$ , respectively; finally, synthesize  $u_x, u_y$ , and  $u_z$  into a value  $u$  as the interpolated value at point  $(x, y, z)$ .

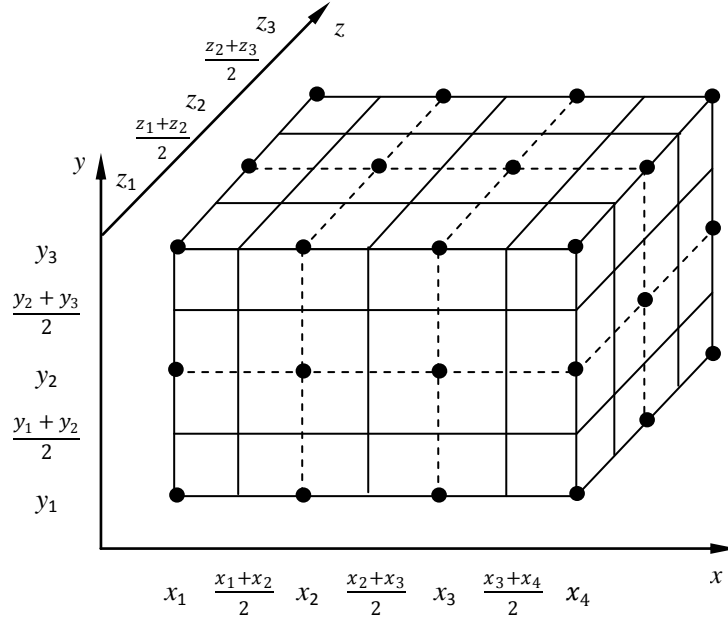


Figure 5-6 An illustration of interpolation regions partitioned by base points and their approximation domains in three-dimensional AD interpolation

Where the regions enclosed by solid lines are the approximation domains of base points  $(x_i, y_j, z_k)$ , and the dotted lines further divide an approximation domain into two, four, or eight parts

There were 4 formulas separated as 2 pairs in the above two-dimensional interpolation, for here three-dimensional interpolation we have 6 formulas separated as 3 pairs:

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i-1,j,k}}{x_i - x_{i-1}} x + \frac{x_i u_{i-1,j,k} - x_{i-1} u_{ijk}}{x_i - x_{i-1}}, & x \in \left[ \frac{x_i + x_{i-1}}{2}, x_i \right] \end{aligned} \right. \quad (5-6)$$

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i+1,j,k}}{x_i - x_{i+1}} x + \frac{x_i u_{i+1,j,k} - x_{i+1} u_{ijk}}{x_i - x_{i+1}}, & x \in \left[ x_i, \frac{x_i + x_{i+1}}{2} \right] \end{aligned} \right. \quad (5-7)$$

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i,j-1,k}}{y_j - y_{j-1}} y + \frac{y_j u_{i,j-1,k} - y_{j-1} u_{ijk}}{y_j - y_{j-1}}, & y \in \left[ \frac{y_j + y_{j-1}}{2}, y_j \right] \end{aligned} \right. \quad (5-8)$$

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i,j+1,k}}{y_j - y_{j+1}} y + \frac{y_j u_{i,j+1,k} - y_{j+1} u_{ijk}}{y_j - y_{j+1}}, & y \in \left[ y_j, \frac{y_j + y_{j+1}}{2} \right] \end{aligned} \right. \quad (5-9)$$

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i,j,k-1}}{z_k - z_{k-1}} z + \frac{z_k u_{i,j,k-1} - z_{k-1} u_{ijk}}{z_k - z_{k-1}}, & z \in \left[ \frac{z_k + z_{k-1}}{2}, z_k \right] \end{aligned} \right. \quad (5-10)$$

$$\left\{ \begin{aligned} u &= \frac{u_{ijk} - u_{i,j,k+1}}{z_k - z_{k+1}} z + \frac{z_k u_{i,j,k+1} - z_{k+1} u_{ijk}}{z_k - z_{k+1}}, & z \in \left[ z_k, \frac{z_k + z_{k+1}}{2} \right] \end{aligned} \right. \quad (5-11)$$

where  $u_{ijk} = f(x_i, y_j, z_k)$ .

Similar to two-dimensional AD interpolation, here for an evaluated point  $(x, y, z) \in U$ , we need only choose a formula with respect to  $x$ , a formula with respect to  $y$ , and a formula with respect to  $z$ . As to which 3 formulas will be used, it should be determined by the subintervals that  $x$ ,  $y$  and  $z$  lie respectively. For example, if  $x \in \left[ \frac{x_i + x_{i-1}}{2}, x_i \right]$ ,  $y \in \left[ y_j, \frac{y_j + y_{j+1}}{2} \right]$ , and  $z \in \left[ \frac{z_k + z_{k-1}}{2}, z_k \right]$ , then Formulas (5-6), (5-9), and (5-10) should be respectively used. By

Formula (3-12) in Section 3.2, here the formula that synthesizes  $u_x$ ,  $u_y$ , and  $u_z$  is

$$u = u_x + u_y + u_z - 2u_{ijk} \quad (5-12)$$

**Example 5.4** Use AD interpolation to do interpolating to function of three variables,  $u = -x^2 - y^2 - z^2$ .

We take sampled-data points,  $((x_i, y_j, z_k), u_{ijk})$ , as follows:

$$x_i: -3, -2.5, -2.0, \dots, -0.5, 0, 0.5, \dots, 2.0, 2.5, 3.$$

$$y_j: -4, -3.5, -3.0, \dots, -0.5, 0, 0.5, \dots, 3.0, 3.5, 4.$$

$$z_k: -4, -3.5, -3.0, \dots, -0.5, 0, 0.5, \dots, 3.0, 3.5, 4.$$

$$u_{ijk}: u = -x_i^2 - y_j^2 - z_k^2$$

and take evaluated points,  $(x, y, z)$  as follows:

$$x: -3, -2.75, -2.50, \dots, -0.75, 0, 0.25, \dots, 2.50, 2.75, 3.$$

$$y: -4, -3.80, -3.60, \dots, -0.80, 0, 0.20, \dots, 3.60, 3.80, 4.$$

$$z: -4, -3.75, -3.50, \dots, -0.75, 0, 0.25, \dots, 3.50, 3.75, 4.$$

Doing AD interpolation (the corresponding  $u$ -values obtained are omitted), the interpolation effect (slice chart) is shown in Figure 5-7.

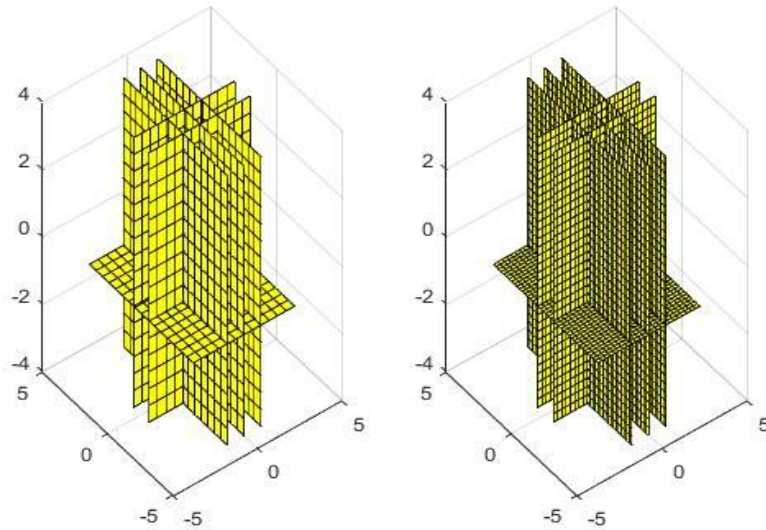


Figure 5-7 The effect drawing (slice chart) of AD interpolation to function of three variables,  $u = -x^2 - y^2 - z^2$

Where the graph on the left is the graph before interpolating, and the graph on the right is the graph after interpolating

#### Accuracy analysis:

maximum error 0.1850, minimum error 0, average error 0.0993

### 5.3 N-Dimensional AD Interpolation

Generalizing the above two-dimensional and three-dimensional AD interpolations, we obtain a general  $n$ -dimensional AD interpolation method.

Suppose some data points of function of  $n$  variables,  $y = f(x_1, x_2, \dots, x_n)$ ,  $((x_{1i}, x_{2j}, \dots, x_{nk}), y_{ijk})$  ( $i=1, 2, \dots, r; j=1, 2, \dots, s; k=1, 2, \dots, t$ ) are known, and the base points

$(x_{1_i}, x_{2_j}, \dots, x_{n_k})$  are regularly distributed, then the procedure of  $n$ -dimensional AD interpolation is:

(1) Partition the  $n$ -dimensional interpolation space  $U=[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] = [x_{1_1}, x_{1_r}] \times [x_{2_1}, x_{2_s}] \times \dots \times [x_{n_1}, x_{n_t}]$  according to the approximation domains of all base points  $\mathbf{x}_{ij\dots k} = (x_{1_i}, x_{2_j}, \dots, x_{n_k})$  ( $i=1, 2, \dots, r; j=1, 2, \dots, s; k=1, 2, \dots, t$ );

(2) For evaluated point  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in U$ , look for the approximation domain it lies (Which is equivalent to seek its nearest base point  $\mathbf{x}_{ij\dots k}$  ( $i \in \{1, 2, \dots, r\}, j \in \{1, 2, \dots, s\}, k \in \{1, 2, \dots, t\}$ ));

(3) Separate the  $n$ -dimensional interpolating computation of point  $\mathbf{x}$  with base point  $\mathbf{x}_{ij\dots k}$  into  $n$  one-dimensional interpolating computations of  $x_1$  with  $x_{1_i}$ ,  $x_2$  with  $x_{2_j}$ ,  $\dots$ , and  $x_n$  with  $x_{n_k}$ , and do interpolating computations with the corresponding one-dimensional interpolation formulas, respectively;

(4) Synthesize the  $n$  got approximate values  $y_{x_1}, y_{x_2}, \dots, y_{x_n}$  into one value  $y$  as the interpolated value at point  $\mathbf{x}$ .

$N$ -dimensional AD interpolation has  $n$  pairs of interpolating formulas as follows:

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{i-1,j\dots k}}{x_{1_i} - x_{1_{i-1}}} x_1 + \frac{x_{1_i} y_{i-1,j\dots k} - x_{1_{i-1}} y_{ij\dots k}}{x_{1_i} - x_{1_{i-1}}}, & x_1 \in [\frac{x_{1_i} + x_{1_{i-1}}}{2}, x_{1_i}] \end{cases} \quad (5-13)$$

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{i+1,j\dots k}}{x_{1_i} - x_{1_{i+1}}} x_1 + \frac{x_{1_i} y_{i+1,j\dots k} - x_{1_{i+1}} y_{ij\dots k}}{x_{1_i} - x_{1_{i+1}}}, & x_1 \in [x_{1_i}, \frac{x_{1_i} + x_{1_{i+1}}}{2}] \end{cases} \quad (5-14)$$

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{ij\dots k-1}}{x_{2_i} - x_{2_{i-1}}} x_2 + \frac{x_{2_i} y_{ij\dots k-1} - x_{2_{i-1}} y_{ij\dots k}}{x_{2_i} - x_{2_{i-1}}}, & x_2 \in [\frac{x_{2_i} + x_{2_{i-1}}}{2}, x_{2_i}] \end{cases} \quad (5-15)$$

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{ij\dots k+1}}{x_{2_i} - x_{2_{i+1}}} x_2 + \frac{x_{2_i} y_{ij\dots k+1} - x_{2_{i+1}} y_{ij\dots k}}{x_{2_i} - x_{2_{i+1}}}, & x_2 \in [x_{2_i}, \frac{x_{2_i} + x_{2_{i+1}}}{2}] \end{cases} \quad (5-16)$$

...

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{ij\dots k-1}}{x_{n_i} - x_{n_{i-1}}} x_n + \frac{x_{n_i} y_{ij\dots k-1} - x_{n_{i-1}} y_{ij\dots k}}{x_{n_i} - x_{n_{i-1}}}, & x_n \in [\frac{x_{n_i} + x_{n_{i-1}}}{2}, x_{n_i}] \end{cases} \quad (5-17)$$

$$\begin{cases} y = \frac{y_{ij\dots k} - y_{ij\dots k+1}}{x_{n_i} - x_{n_{i+1}}} x_n + \frac{x_{n_i} y_{ij\dots k+1} - x_{n_{i+1}} y_{ij\dots k}}{x_{n_i} - x_{n_{i+1}}}, & x_n \in [x_{n_i}, \frac{x_{n_i} + x_{n_{i+1}}}{2}] \end{cases} \quad (5-18)$$

where  $y_{ij\dots k} = f(x_{1_i}, x_{2_j}, \dots, x_{n_k})$ .

And the synthesizing formula of the final interpolated value is

$$y = \sum_{i=1}^n y_{x_i} - (n-1) y_{ij\dots k} \quad (5-19)$$

Here  $y_{x_i}$  are the approximate values that got by one-dimensional AD interpolation computations of  $x_i$  ( $i=1, 2, \dots, n$ ).

The Formula (5-19) is the **Sum-Times-Difference formulas** in AD interpolation.

It can be seen that in theory the dimensionality  $n$  of  $n$ -dimensional AD interpolation has no an upper limit. In this way, using the technique of “first separating then synthesizing” and the Sum-Times-Difference formula, we can also realize the high-dimensional or even super-high-dimensional AD interpolations with more than 3 dimensions. In fact, a program



for AD interpolation with more than 10 dimensions has been completed by the author, and the dimension can be easily increased (due to the space limit, the relevant content will be introduced in another article).

Actually, if we substitute the interpolation formulas into the Sum-Times-Difference formula, it is not difficult to see that the Sum-Times-Difference formula is actually a linear combination of all coordinate components of point  $(x_1, x_2, \dots, x_n)$ . For example, we substitute two-dimensional interpolating formulas  $z = \frac{z_{ij} - z_{i-1,j}}{x_i - x_{i-1}}x + \frac{x_i z_{i-1,j} - x_{i-1} z_{ij}}{x_i - x_{i-1}}$  and  $z = \frac{z_{ij} - z_{i,j-1}}{y_j - y_{j-1}}y + \frac{y_j z_{i,j-1} - y_{j-1} z_{ij}}{y_j - y_{j-1}}$  into corresponding Sum-Times-Difference formula  $z = z_x + z_y - z_{ij}$ , then it follows that

$$z = \frac{z_{ij} - z_{i-1,j}}{x_i - x_{i-1}}x + \frac{z_{ij} - z_{i,j-1}}{y_j - y_{j-1}}y + \left( \frac{x_i z_{i-1,j} - x_{i-1} z_{ij}}{x_i - x_{i-1}} + \frac{y_j z_{i,j-1} - y_{j-1} z_{ij}}{y_j - y_{j-1}} - z_{ij} \right)$$

Therefore, Sum-Times-Difference formula can also be said to be the interpolation formula of multidimensional AD interpolation.

Although AD interpolation is also a local linear interpolation, it is derived on the basis of approximate evaluation of functions based on the approximation-degree, so the base points involved in interpolation are related to the position of the evaluated point  $\mathbf{x} \in U$  relative to its nearest base point  $\mathbf{x}_{i,j,\dots,k}$ . It can be seen that  $n$ -dimensional AD interpolation involves a total of  $1+n$  base points, and when dimensionality is added by 1, the number of the base point involved will be added by only 1 (see Figures 3-7, 3-8 and 5-1, 5-5). Because the point  $\mathbf{x}$  is only approximate to the base point  $\mathbf{x}_{i,j,\dots,k}$ , so the  $n$  approximate values,  $y_{x_1}, y_{x_2}, \dots, y_{x_n}$ , obtained from one-dimensional interpolating computations got from separating are most affected by base point  $\mathbf{x}_{i,j,\dots,k}$ , and final interpolated value  $y$  is also most affected by  $\mathbf{x}_{i,j,\dots,k}$ . In other words, AD interpolation is dominated by the nearest base point  $\mathbf{x}_{i,j,\dots,k}$  of evaluated point  $\mathbf{x}$ .

## 6 Accuracy and Efficiency about AD Interpolation

### 6.1 The Accuracy

It can be seen from the one-dimensional interpolation formulas and the Sum-Times-Difference formula of multi-dimensional interpolation that AD interpolation is essentially a piecewise (block) linear interpolation. Therefore, the accuracy of AD interpolation should be on the level of linear interpolation. In fact, the data experiments show that: the accuracy of 1, 2, 3-D AD interpolations are about the same as the traditional linear interpolation, especially for the quadratic functions (such as previous examples 1, 2, 3), the accuracy of the two is exactly the same. In addition, we find that to the interpolated function  $y=f(x_1, x_2, \dots, x_n)$  which is monotonous on each interpolation subinterval, in theory, the maximum error of AD interpolation is

$$E = \left| f\left(\frac{x_{1_i} + x_{1_{i+1}}}{2}, \frac{x_{2_i} + x_{2_{i+1}}}{2}, \dots, \frac{x_{n_i} + x_{n_{i+1}}}{2}\right) - \hat{f}\left(\frac{x_{1_i} + x_{1_{i+1}}}{2}, \frac{x_{2_i} + x_{2_{i+1}}}{2}, \dots, \frac{x_{n_i} + x_{n_{i+1}}}{2}\right) \right|$$

Here,  $f(\mathbf{x})$  and  $\hat{f}(\mathbf{x})$  are respectively the interpolated value and desired value at point  $\mathbf{x}=(x_1, x_2, \dots, x_n)$ .

Actually, from the principle of AD interpolation, it is not difficult to see that the accuracy of AD interpolation is related to two factors: one is the size of the approximation domain of correlative base point and the other is the approximation-degree of the evaluated point  $x$  to the corresponding nearest base point. Obviously, the narrower the approximation domain (including the relative approximation domain and the absolute approximation domain) is, the higher the interpolation accuracy is; And when an approximation domain has be determined, the higher the approximation-degree of the point  $x$  to the corresponding base point is, the higher the accuracy of the corresponding interpolated value is.

Thus, for the regularly distributed base points, as long as there are enough data points, that is, enough samples, the required any accuracy can be achieved by using AD interpolation.

## 6.2 The Efficiency

Viewed from the principle, the difference between AD interpolation and traditional linear interpolation is mainly in multidimensional interpolations. In fact, in multidimensional AD interpolation, by using a technique of "first separating and then synthesizing", an  $n$ -dimensional AD interpolation computation is separated into  $n$  one-dimensional AD interpolation computations to be done respectively; then the got results are synthesized into a value by Sum-Times-Difference formula as the result value of the  $n$ -dimensional interpolation. In this way, one-time  $n$ -dimensional AD interpolation computation becomes into  $n$ -times one-dimensional AD interpolation computations and one-time synthesizing computation. From the Sum-Times-Difference formula, all of multidimensional AD interpolations have only two layers of computation: one-dimensional AD interpolation computations and synthesizing computation (as shown in Figure 6-1 (a)), and when dimensionality is added by 1, only one-time one-dimensional AD interpolation computation and one-time addition operation are added in the corresponding AD interpolation (as shown in Figure 6-1 (b)). That is to say, the two-layer computing structure of AD interpolation is always unchanged, but the number of terms of computation in lower layer will increase with the increase of data dimension. In this way, the times of the formula (including interpolation formulas and Sum-Times-Difference formula) computation in  $n(n>1)$ -dimensional AD interpolation is  $n+1$ . Starting from the two dimensions, the times of formula computation in the AD interpolation are in order: 3, 4, 5, ..., i.e., a sequence of equal difference numbers with a difference of 1, and the general term is:  $n+1$ . We denote the times of the formula computation in  $n(n>1)$ -dimensional AD interpolation as  $N_n$ , and the recurrence formula is:  $N_{n+1} = N_n + 1$ . From this we see that the computation load of multidimensional AD interpolation is much smaller than that of traditional multidimensional linear interpolation.

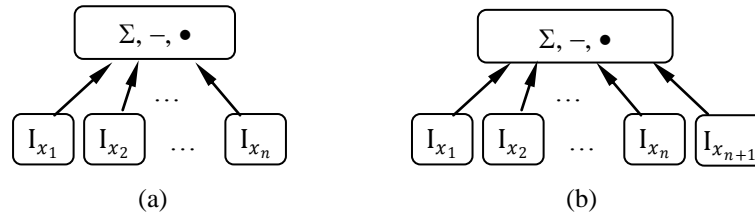


Figure 6-1 The hierarchical structure of formula computation in multidimensional AD interpolation

Where  $I_{x_i}$  denote the one-dimensional AD interpolating computations of variable  $x_i$  ( $i=1,2,\dots,n,n+1$ )

In addition, since  $n$  one-dimensional AD interpolation computations got from separation are independent of each other, so the parallel computation can be used. Thus, the efficiency of  $n$ -dimensional AD interpolation is almost equal to that of one-dimensional AD interpolation

From the above mentioned, we can see that the high efficiency of AD interpolation is obvious, and the higher the dimension is, the higher the efficiency is.

## 7 Summary and Prospect

Inspired by the approximate evaluation method of flexible linguistic functions in reference [1], in this paper we introduce the measure of approximate-degree and the concept of approximate-degree function between numerical values, and according to "approximation axiom" and by computing approximation-degree, transferring approximation-degree, evaluating inverse function, and choosing approximate value, we realize the approximate evaluation of a numerical function; Especially, for multivariate functions, in the sense of "strict approximation", we use the technique of "first separating then synthesizing" and find and present a formula of "Sum-Times-Difference", and achieve corresponding approximate evaluation more skillfully. On this basis, we further derive a set of interpolation formulas for regularly distributed base points, thus an interpolation method based on approximation-degree, approximation-degree-based interpolation, i.e., AD interpolation, has been developed; and then, we extend one-dimensional AD interpolation to high-dimensional interpolation. As a test of AD interpolation method, we apply AD interpolation to the actual functions and obtain satisfactory results: the one-dimensional AD interpolation has the effect of "reaching the same goal from different routes" with usual piecewise linear interpolation, the two-dimensional and three-dimensional AD interpolations are comparable to the usual bilinear and trilinear interpolations; and then, the high-dimensional AD interpolations with more than 3 dimensions also have passed the data test.

Actually, AD interpolation is also a kind of linear interpolation in essence, its accuracy is about the same as that of traditional (piecewise) linear interpolation, but its principle is different from that of the latter. One-dimensional AD interpolation is done directly by using corresponding interpolation formulas;  $n(n>1)$ -dimensional AD interpolation is firstly separated into  $n$  one-dimensional AD interpolation computations which can be processed in parallel to do respectively, and then the results got are synthesized by Sum-Times-Difference formula into a value as the result value of the  $n$ -dimensional interpolation. AD interpolation is dominated by a nearest base point and assisted by adjacent base point(s) of the nearest base point,  $n$ -dimensional AD interpolation computation involves  $1+n$  base points, which means that when dimensionality is added by 1, only one correlative base point will be added. The  $n$ -dimensional AD interpolation has two layers of computation: first, parallel one-dimensional AD interpolation computations, then, synthesizing computation, and when dimensionality is added by 1, only one term of one-dimensional AD interpolation computation and one-time addition operation are added. In this way, compared with the traditional linear interpolation, the computation load of the AD interpolation is much smaller, if the parallel processing is also used, the efficiency of  $n$ -dimensional AD interpolation is almost the same as that of the

one-dimensional AD interpolation. In addition, the dimensionality of the multi-dimensional AD interpolation is theoretically unlimited (limited only by the associated computational resources).

In a word, AD interpolation has the advantages of novel idea, unique method, simple computation, adjustable accuracy, high efficiency, and unlimited dimension. This opens up a new way for data fitting and function approximation, and especially, it starts a feasible and convenient approach and provides an effective method for high-dimensional and super-high-dimensional interpolations.

This paper expounds the basic principle of AD interpolation. On the basis, we can further carry out the following work:

- Applying AD interpolation to practical high-dimensional and super-high dimensional interpolation problems

- AD interpolation with irregular distributed base points
- AD interpolation for vector-valued functions
- AD interpolation with derivative (partial derivative and directional derivative)
- Using AD interpolation to realize a function approximator

It can be seen that the approximator using AD interpolation is always interpretable, and won't occur "the dilemma between precision and interpretability"<sup>[7]</sup> encountered by the function approximator based on fuzzy technology.

- Introducing AD interpolation into machine learning to realize a new instance-based learning method——AD interpolation learning

Actually, for these further topics, the author has done some research and obtained some preliminary results. Due to space constraints, we will discuss them separately.

## References

- [1] Shiyu Lian, 2016. Principles of Imprecise-Information Processing: A New Theoretical and Technological System, Springer Nature.
- [2] David Kincaid and Ward Cheney, 2002. Numerical Analysis: Mathematics of Scientific Computing (Third Edition), Thomson Learning.
- [3] Tom M. Mitchell, 1997. Machine Learning, McGraw-Hill Companies, Inc.
- [4] Ethem Alpaydin, 2014. Introduction to Machine Learning (Third Edition), Massachusetts Institute of Technology, MIT Press
- [5] Shiyu Lian, 2016. Correspondence between Flexible Sets, and Flexible Linguistic Functions, in: Shiyu Lian, Principles of Imprecise-Information Processing: A New Theoretical and Technological System, Springer Nature, pp. 205–228.
- [6] Shiyu Lian, 2016. Approximate Evaluation of Flexible Linguistic Functions, in: Shiyu Lian, Principles of Imprecise-Information Processing: A New Theoretical and Technological System, Springer Nature, pp. 393–417.
- [7] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, 1997. Neuro-Fuzzy and Soft Computing (Prentice Hall, Upper Saddle River, NJ), pp. 342~245, 382~385.